 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--


DOWNLOADED AND/OR HARD COPY UNCONTROLLED

Verify that this is the correct version before use.

APPROVAL SIGNATURES		DATE
Gregory Blaney (original signature on file)	Management System Representative	05/13/2004

REVISION HISTORY			
Rev. No.	Description of Change	Author	Effective Date
Basic	Initial Release	Bill Jackson IT/215	08/15/97
A	Pages 4 through 9 changed to reflect IV&V purchased, DCR-9A review comments incorporated	Bill Jackson IT/215	03/06/98
B	Ames SLP format	Bill Jackson IT/215	04/29/98
C	Quality Record - format changes	Bill Jackson IT/215	08/26/98
D	Section 6.3 is modified to include the technical report for design and verification reviews. Section 8.0 is modified to add technical report to quality records	Bill/Jackson IT/215	10/08/98
E	References to Ames Quality Manual replaced with references to IV&V Facility Quality Manual	Bill Jackson IT/215	09/10/99
F	Format and Number changes; Delete Reference to Ames Research Center	Griggs	12/01/00
G	Modified to correct Quality Record identification	Costello	04/16/01
H	Modified to incorporate WBS resulting from IV&V Transition	Dan Solomon	05/14/04

CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
 VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

REFERENCE DOCUMENTS	
Document Number	Document Title
NPR 1441.1	NASA Records Retention Schedule
IVV 05	Document and Data Control
IVV 09-4	Project Management
IVV 16	Control of Quality Records

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

1.0 Purpose

The purpose of this system level procedure (SLP) is to establish a consistent method for providing IV&V technical services to customers, sufficient to ensure safety and risk mitigation for the successful deployment of software-intensive systems.

2.0 Scope

This procedure is applicable to IV&V technical activities provided by the NASA IV&V Facility.

3.0 Definitions and Acronyms

3.1 Verification

The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase.

3.2 Validation

The process of evaluating software at the end of its software development process to ensure compliance with software requirements. This process ensures that the software system performs to the customer's expectations under operational conditions.


3.3 NASA Project Manager

IV&V Facility civil servant personnel, appointed by IV&V Facility Management, who perform project management functions.

3.4 Analyst

The Civil Service or contractor person assigned to perform a specific IV&V task.

3.5 Project

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

For the context of this SLP, a project is one or more IV&V, IA, or System/Software-Engineering task(s) being performed by the IV&V Facility for a specific customer.

3.6 Acceptance Testing

Testing conducted in an operational environment to determine whether a system satisfies its acceptance criteria (i.e., initial requirements and current needs of its user) and to enable the customer to determine whether to accept the system.

3.7 System Testing

An orderly progression of testing of incremental pieces of the system in which both hardware and software elements, are combined and tested until the entire system has been integrated to verify the requirements of the system and to validate whether the system meets its original objectives.

3.8 Software Final Qualification Testing (FQT)


Final testing of the complete software program to verify that the software meets all of the software requirements and is ready to be integrated with system hardware.

3.9 Software Integration Testing

An orderly progression of testing of incremental pieces of the software program in which software elements are combined and tested to show compliance with the software design, capabilities, and requirements. Testing is also performed to verify operation under off-nominal and stress conditions.

3.10 Component Testing

Testing conducted to verify the correct implementation of the design and compliance with program requirements for one software element (e.g., unit, module, CSC) or a collection of software elements (e.g., CSCI).

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

3.11 Test Plan

Documentation that specifies the scope, approach, resources, and schedule of intended test activities.

3.12 Test Design

Documentation that specifies the details of the test approach for a software feature or combination of software features and identifying the associated tests.

3.13 Test Case


Documentation that specifies inputs, predicted results, and a set of execution conditions for a test item.

3.14 Test Procedure

Documentation that specifies a sequence of actions for the execution of a test.

3.15 Acronyms

IV&V	Independent Verification and Validation
ATP	Acceptance Test Plan
CDR	Critical Design Review
CPU	Central Processing Unit
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
FQT	Final Qualification Test
IDD	Interface Design Document
IRS	Interface Requirement Specification
IVVP	Independent Verification and Validation Plan
PDR	Preliminary Design Review
RFP	Request For Proposal
SDD	System Design Document
SDP	Software Development Process
SRS	Software Requirements Specification
SVVP	Software Verification and Validation Plan

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

STP System Test Plan

4.0 Flow Chart

N/A.

5.0 Responsibilities

The responsibilities are defined in Section 6, Procedure.


6.0 Procedure

The Project Manager responsibilities are as defined in IVV 09-4, Project Management.


The IV&V analysis tasks for an individual project shall be defined in the associated project plan (ref: IVV 09-4, Project Management). The following IV&V analysis tasks are typically performed.

Name	WBS	Description
Phase Independent Support 1.0		
Management and Planning of Independent Verification and Validation	1.1	<p>Generate an Independent Verification and Validation Plan (IVVP) for all life cycle processes. The IVVP template should be used when developing this plan. The IVVP may require updating throughout the life cycle. Outputs of several other activities are inputs to the IVVP and the IVVP should be updated when these inputs become available. Establish a baseline IVVP prior to the start of the Planning and Scoping activities. Identify project milestones in the IVVP. Schedule IV&V tasks to support project management reviews and technical reviews. Plan the interface among the IV&V effort, the Project office, and the software developer. Document the data exchange requirements in the IVVP.</p> <p>Plan the IV&V schedule for each IV&V task. Identify the preliminary list of development processes and products to be evaluated by the IV&V process. Describe IV&V access rights to proprietary and classified information and the process to secure and document those rights. The plan should be coordinated with the developer of the software, which may include not only NASA, but also developers contracted to NASA. Incorporate the project software integrity level scheme into the planning process.</p> <p>Review and summarize the IV&V effort to define changes to IV&V tasks or to</p>


CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--


		<p>redirect the IV&V effort. Recommend whether to proceed to the next set of IV&V and development life cycle activities, and provide task reports, anomaly reports, and IV&V Activity Summary Reports to the organization(s) identified in the IVVP.</p> <p>Verify that all IV&V tasks comply with task requirements defined in the IVVP and/or Task Order. Verify that IV&V task results have a basis of evidence supporting the results. Assess all IV&V results and provide recommendations for program acceptance and certification as input to the IV&V Final Report. The management review of IV&V may use any review methodology such as provided in IEEE Std 1028 - 1998.</p> <p>Evaluate proposed changes to the project (e.g., anomaly corrections and requirements changes) for effects on previously completed IV&V tasks and future IV&V tasks. Verify that the change is consistent with system requirements and does not adversely affect other requirements directly or indirectly. An adverse effect is a change that could create new system hazards and risks or impact previously resolved hazards and risks. Plan iteration of affected tasks or initiate new tasks to address the software change or iterative development process.</p> <p>Coordinate the IV&V effort with organizational (e.g., management, improvement) and supporting processes (e.g., quality assurance, joint review, and problem resolution).</p> <p>Use the results of reviews to identify process improvement opportunities in the conduct of IV&V</p>
Issue and Risk Tracking	1.2	Track IV&V generated issues from initiation through closure by the project and/or IV&V Team. Identify and track IV&V project risks and developmental project risks. Provide recommendations to mitigate these risks. Communicate the issues and risks to the appropriate IV&V and development Project management.
Final Report Generation	1.3	Summarize in the IV&V final report(s) the IV&V activities, tasks and results, including status and disposition of anomalies and risks. Provide an assessment of the overall software quality, lessons learned and recommendations for future projects.

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--


IV&V Tool Support	1.4	<p>Plan the tools needed to support the IVV effort. The plan includes a description of each tool's performance, required inputs, outputs generated, need date, and cost of tool purchase or development. The tool plan should also describe test facilities and integration and system laboratories supporting the IVV effort. The scope and rigor of the IVV effort as defined by the selected software integrity level should be considered in defining the performance required of each tool.</p> <p>Procure, develop or tailor software tools in support of the IV&V tasks. Provide training to IV&V personnel as required and provide maintenance and operations support (help desk) for tool use.</p>
Management and Technical Review Support	1.5	<p>Support project management reviews and technical reviews (e.g., PDR and CDR) by assessing the review materials, attending the reviews, presenting at the reviews and providing task reports and anomaly reports (when appropriate). The management and technical review support may also use any review methodology such as provided in IEEE Std 1028 - 1998.</p>
Criticality Analysis	1.6	<p>Develop a software integrity level scheme for the project under consideration using the available documentation (e.g., system requirements, subsystem requirements, software specifications, hazard analyses, etc) and assess the software components against the SIL scheme. Document the software integrity level assigned to individual software components (e.g., requirements, detailed functions, software modules, subsystems, or other software partitions).</p> <p>For IV&V planning purposes, the most critical software integrity level assigned to individual elements shall be the integrity level assigned to the entire software. Verify whether any software component can influence individual software components assigned a higher software integrity level, and if such conditions exist, then assign that software component the same higher software integrity level. Verify whether any software component controls or mitigates a hazardous event and that the proper software integrity level is assigned to those components.</p>

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Identify Process Improvement Opportunities in the Conduct of IV&V	1.7	Gather and analyze lessons learned, risks identified, and IV&V metrics on a periodic basis. Identify any deficiencies in the IV&V process and implement appropriate corrective action(s). Monitor the effectiveness of the corrective actions being taken. Determine if the corrective actions should be incorporated into the IV&V processes. Document findings in appropriate technical reports.
Concept Phase 2.0		
Reuse Analysis	2.1	Analyze the developer's documentation to verify that the original domain of the candidate reuse software will satisfy the domain of the new system (e.g. software integrity level, user needs, operating environment, safety, security, and interfaces). If the developer has performed no domain analysis, perform domain analysis (see IEEE Std 1517-1999) to compare the original domain and the new domain of the candidate reuse software. Verify that developer reuse planning dispositions and documents all domain differences.
System Architecture Assessment	2.2	Assess the proposed architectural schema for feasibility. Assess how the proposed architecture satisfies the users needs in terms of the user's requirements. Examples of requirements are timing, storage, usability, safety, security, and suitability for mission.
System Requirements Review	2.3	Analyze the system requirements (e.g., system requirements specification, feasibility study report, business rules description to: <ol style="list-style-type: none"> 1) Verify the consistency of requirements to user needs, 2) Validate whether the requirements can be satisfied by the defined technologies, methods, and algorithms defined for the project (feasibility), and 3) Verify whether objective information that can be demonstrated by testing is provided in the requirements (testability). Review other requirements such as deliverable definitions, listing of appropriate compliance standards and regulations, user needs, etc. for completeness, correctness, and accuracy.

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Concept Document Evaluation	2.4	<p>Verify that the concept documentation satisfies user needs and is consistent with acquisition needs. Validate constraints of interfacing systems and constraints or limitations of proposed approaches.</p> <p>Analyze concept documents to validate that the following satisfy user needs:</p> <ol style="list-style-type: none"> 1) System functions 2) End-to-end system performance; 3) Operation and maintenance requirements; and 4) Migration requirements from an existing system, where applicable.
Software/User Requirements Allocation Analysis	2.5	<p>Verify the correctness, accuracy, and completeness of the concept requirement allocation to software and user interfaces against user needs.</p> <p>Correctness - Verify that performance requirements (e.g., timing, response time, and throughput) allocated to software and user interfaces satisfy user needs.</p> <p>Accuracy - Verify that the internal and external interfaces specify the data formats, interface protocols, frequency of data exchange at each interface, and other key performance requirements to demonstrate compliance with user requirements.</p> <p>Completeness</p> <ol style="list-style-type: none"> a. Verify that application specific requirements such as functional diversity, fault detection, fault isolation, and diagnostic and error recovery satisfy user needs b. Verify that the user's maintenance requirements for the system are completely specified c. Verify that the migration from the existing system and replacement of the system satisfy user needs
Traceability Analysis - System	2.6	<p>Identify all system requirements that will be implemented completely or partially by software. Verify that these system requirements are traceable to user or program requirements. Start the software requirements traceability analysis with system requirements.</p>

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--


Requirements Phase 3.0

Traceability Analysis - Requirements	3.1	<p>Trace the software requirements (SRS and IRS) to system requirements (Concept Documentation), and system requirements to the software requirements. Analyze identified relationships for correctness, consistency, completeness, and accuracy.</p> <p>Correctness - Validate that the relationships between each software requirement and its system requirement are correct</p> <p>Consistency - Verify that the relationship between the software and system requirements are specified to a consistent level of detail</p> <p>Completeness</p> <ul style="list-style-type: none"> a. Verify that every software requirement is traceable to a system requirement with sufficient detail to show compliance with the system requirement b. Verify that all system requirements related to software are traceable to software requirements <p>Accuracy - Validate that the system performance and operating characteristics are accurately specified by the traced software requirements</p>
Software Requirements Evaluation	3.2	<p>Evaluate the requirements (e.g., functional, capability, interface, qualification, safety, security, human factors, data definitions, user documentation, installation and acceptance, user operation and user maintenance) of the SRS and IRS for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are as follows:</p> <p>Correctness</p> <ul style="list-style-type: none"> a. Verify and validate that the software requirements satisfy the system requirements allocated to software within the assumptions and constraints of the system. b. Verify that the software requirements comply with applicable standards, references, regulations, policies, physical laws, and business rules. c. Validate the sequence of states and state changes. d. Validate that the flow of data and control satisfy functionality and performance requirements. e. Validate data usage and format. <p>Consistency</p> <ul style="list-style-type: none"> a. Verify that all terms and concepts are documented consistently b. Verify that the function interactions and assumptions are consistent and satisfy system requirements and acquisition needs. c. Verify that there is internal consistency between the software requirements and external consistency with the system requirements <p>Completeness</p> <ul style="list-style-type: none"> a. Verify that the following elements are in the SRS or IRS within the assumptions and constraints of the system: <ul style="list-style-type: none"> 1. Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging; 2. Process definition and scheduling;

CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--


		<ol style="list-style-type: none"> 3. Hardware, software, and user interface descriptions; 4. Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety and security); 5. Critical configuration data; and 6. System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing). <ol style="list-style-type: none"> b. Verify that the SRS and IRS satisfy specified configuration management procedures. <p>Accuracy</p> <ol style="list-style-type: none"> a. Validate that the logic, computational, and interface precision (e.g., truncation and rounding) satisfy the requirements in the system environment. b. Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws <p>Readability</p> <ol style="list-style-type: none"> a. Verify that the documentation is legible, understandable, and unambiguous to the intended audience. b. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms and symbols <p>Testability</p> <p>Verify that there are objective acceptance criteria for validating the requirements of the SRS and IRS</p>
Interface Analysis - Requirements	3.3	<p>Verify and validate that the requirements for software interfaces with hardware, user, operator, and other systems are correct, consistent, complete, accurate, and testable. The task criteria are as follows:</p> <p>Correctness - Validate the external and internal system and software interface requirements.</p> <p>Consistency - Verify that the interface descriptions are consistent between the SRS and IRS.</p> <p>Completeness - Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety and security).</p> <p>Accuracy - Verify that each interface provides information with the required accuracy.</p> <p>Testability - Verify that there are objective acceptance criteria for validating the interface requirements</p>
System Test Plan Analysis	3.4	<p>Verify that the System Test plan conforms to project defined test document purpose, format and content. Validate that the System Test Plan satisfies the following criteria:</p> <ol style="list-style-type: none"> 1) Test coverage of system requirements; 2) Appropriateness of test methods and standards used; 3) Feasibility of system qualification testing; and 4) Feasibility and testability of operation and maintenance requirements.

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--


Acceptance Test Plan Analysis	3.5	<p>Verify that the Acceptance Test Plan complies with project defined test document purpose, format, and content. Validate that the Acceptance Test Plan satisfies the following criteria:</p> <ol style="list-style-type: none"> 1) Test coverage of system requirements; and 2) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs).
Timing and Sizing Analysis	3.6	<p>Collect and analyze data about the software functions and resource utilizations to determine if system and software requirements for speed and capacity are satisfied. The types of functions and resource utilization issues include, but are not limited to the following:</p> <ol style="list-style-type: none"> 1) CPU Load 2) Random access memory and secondary storage 3) Network speed and capacity 4) Input and output speed <p>Sizing and timing analysis is started at software design and iterated through acceptance testing.</p>

Design Phase 4.0


Traceability Analysis - Design	4.1	<p>Analyze trace relationships between design elements (SDD and IDD) and requirements (SRS and IRS) for correctness, consistency, and completeness. The task criteria are as follows:</p> <p>Correctness - Validate the relationship between each design element and the software requirement.</p> <p>Consistency - Verify that the relationship between the design elements and the software requirements are specified to a constant level of detail.</p> <p>Completeness –</p> <ol style="list-style-type: none"> a. Verify that all design elements are traceable from the software requirements. b. Verify that all software requirements are traceable to the design elements.
--------------------------------------	-----	---

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Software Design Evaluation	4.2	<p>Evaluate the design elements (SDD and IDD) for correctness, consistency, completeness, accuracy, readability and testability. The task criteria are as follows:</p> <p>Correctness - Verify and validate that the software design satisfies the software requirements. Verify that the software design complies with applicable standards, references, regulations, policies, physical laws, and business rules. Validate the software design sequences of states and state changes using logic and data flows coupled with domain expertise, prototyping results, engineering principles, or other basis. Validate that the flow of data and control satisfy functionality and performance requirements. Validate data usage and format. Assess the appropriateness of design methods and standards.</p> <p>Consistency - Verify that all terms and design concepts are documented consistently. Verify that there is internal consistency between the design elements and external consistency with architectural design.</p> <p>Completeness - Verify that the following elements are in the SDD, within the assumptions and constraints of the system:</p> <ol style="list-style-type: none"> 1) Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging; 2) Process definition and scheduling; 3) Hardware, software, and user interface descriptions; 4) Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety and security); 5) Critical configuration data; and 6) System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing). 7) Verify that the SDD and IDD satisfy specified configuration management procedures. <p>Accuracy - Validate that the logic, computational, and interface precision (e.g., truncation and rounding) satisfy the requirements in the system environment. Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws.</p> <p>Readability - Verify that the documentation is legible, understandable, and unambiguous to the intended audience. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, symbols, and design language, if any.</p> <p>Testability - Verify that there are objective acceptance criteria for validating each software design element and the system design. Verify that each software design element is testable to objective acceptance criteria.</p>
----------------------------------	-----	---


 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Interface Analysis - Design	4.3	<p>Verify and validate that the software design interfaces with hardware, user, operator, software, and other systems for correctness, consistency, completeness, accuracy, and testability. The task criteria are as follows:</p> <p>Correctness - Validate the external and internal software interface design in the context of the system requirements.</p> <p>Consistency - Verify that the interface design is consistent between the SDD and IDD.</p> <p>Completeness - Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security).</p> <p>Accuracy - Verify that each interface provides information with the required accuracy.</p> <p>Testability - Verify that there are objective acceptance criteria for validating the interface design.</p>
Software FQT Plan Analysis	4.4	<p>Verify that the Final Qualification Test Plan complies with project defined test document purpose, format, and content. Validate that the FQT Plan satisfies the following criteria:</p> <ol style="list-style-type: none"> 1) Traceable to the software requirements; 2) External consistency with the software requirements; 3) Internal consistency; 4) Test coverage of the software requirements; 5) Appropriateness of test standards and methods used; 6) Feasibility of software qualification testing; and 7) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs).
Software Integration Test Plan Analysis	4.5	<p>Verify that the Integration Test Plan complies with project defined test document purpose, format, and content. Validate that the Integration Test Plan satisfies the following criteria:</p> <ol style="list-style-type: none"> 1) Compliance with increasingly larger set of functional requirements at each stage of integration; 2) Assessment of timing, sizing, and accuracy; 3) Performance at boundaries and under stress conditions; and 4) Traceable to the system requirements; 5) External consistency with the system requirements; 6) Internal consistency; 7) Test coverage of the software requirements; 8) Measures of requirements test coverage and software reliability. 9) Appropriateness of test standards and methods used; 10) Feasibility of software qualification testing; and 11) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs).

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Database Analysis	4.6	<p>Evaluate database design including the following:</p> <ol style="list-style-type: none"> 1) Physical Limitations Analysis. Identify the physical limitations of the database such as maximum number of records, maximum record length, largest numeric value, smallest numeric value, and maximum array length in a data structure and compare them to designed values. 2) Index vs. Storage Analysis. Analyze the use of multiple indexes compared to the volume of stored data to determine if the proposed approach meets the requirements for data retrieval performance and size constraints. 3) Data Structures Analysis. Some database management systems have specific data structures within a record such as arrays, tables, and date formats. Review the use of these structures for potential impact on requirements for data storage and retrieval. 4) Backup and Disaster Recovery Analysis. Review the methods employed for backup against the requirements for data recovery and system disaster recovery and identify deficiencies.
Component Test Plan Analysis	4.7	<p>Verify that the Component Test Plan complies with project defined test document purpose, format, and content. Validate that the Component Test Plan satisfies the following criteria:</p> <ol style="list-style-type: none"> 1) Compliance with design requirements; 2) Assessment of timing, sizing, and accuracy; 3) Performance at boundaries and interfaces and under stress and error conditions; and 4) Traceable to the software requirements and design; 5) External consistency with the software requirements and design; 6) Internal consistency between unit requirements; 7) Test coverage of requirements in each unit; 8) Measures of requirements test coverage and software reliability and maintainability 9) Feasibility of software integration and testing; and 10) Feasibility of operation and maintenance (e.g., capability to be operated and maintained in accordance with user needs).
Data Flow Analysis	4.8	<p>Evaluate data flow diagrams including the following:</p> <ol style="list-style-type: none"> 1) Symbology Consistency Check. Verify that each symbol is used consistently. 2) Flow Balancing. Compare the output data from each process block to the data inputs and the data derived within the process to ensure the data is available when required. This process does not specifically examine timing or sequence considerations. 3) Confirmation of Derived Data. Examine the data derived within a process for correctness and format. Data designed to be entered into a process by operator action should be confirmed to ensure availability. 4) Keys to Index Comparison. Compare the data keys used to retrieve data from data stores within a process to the database index design to confirm that no invalid keys have been used and the uniqueness properties are consistent.


CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Implementation Phase 5.0


Traceability Analysis - Code	5.1	<p>Trace the source code components to the corresponding design specification(s), and design specification(s) to source code components. Analyze identified relationships for correctness, consistency, and completeness. The task criteria are as follows:</p> <p>Correctness - Validate the relationship between the source code components and design elements(s).</p> <p>Consistency - Verify that the relationship between source code components and design elements are specified to a consistent level of detail.</p> <p>Completeness</p> <ol style="list-style-type: none"> Verify that all source code components are traceable from the design elements Verify that all design elements are traceable to the source code components
Source Code and Documentation Evaluation	5.2	<p>Evaluate the source code components (Source Code Documentation) for correctness, consistency, completeness, accuracy, readability, and testability. The task criteria are as follows:</p> <p>Correctness</p> <ol style="list-style-type: none"> Verify and Validate that the source code component satisfies the software design Verify that the source code components comply with applicable standards, references, regulations, policies, physical laws, and business rules Validate the source code component sequences of states and state changes Validate that the flow of data and control satisfy functionality and performance requirements Validate data usage and format Assess the appropriateness of coding methods and standards <p>Consistency</p> <ol style="list-style-type: none"> Verify that all terms and code concepts are documented consistently Verify that there is internal consistency between the source code components Validate external consistency with the software design and requirements <p>Completeness</p> <ol style="list-style-type: none"> Verify that the following elements are in the source code, within the assumptions and constraints of the system: <ol style="list-style-type: none"> Functionality (e.g., algorithms, state/mode definitions, input/output validation, exception handling, reporting and logging; Process definition and scheduling; Hardware, software, and user interface descriptions; Performance criteria (e.g., timing, sizing, speed, capacity, accuracy, precision, safety and security); Critical configuration data;

CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

		<p>6. System, device, and software control (e.g., initialization, transaction and state monitoring, and self-testing).</p> <p>b. Verify that the source code documentation satisfies specified configuration management procedures</p> <p>Accuracy</p> <p>a. Validate the logic, computation, and interface precision (e.g., truncation and rounding) in the system environment</p> <p>b. Validate that the modeled physical phenomena conform to system accuracy requirements and physical laws</p> <p>Readability</p> <p>a. Verify that the documentation is legible, understandable, and unambiguous to the intended audience</p> <p>b. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms and symbols</p> <p>Testability</p> <p>a. Verify that there are objective acceptance criteria for validating each source code component</p> <p>b. Verify that each source code component is testable against objective acceptance</p>
Interface Analysis - Code	5.3	<p>Verify and validate that the software source code interfaces with hardware, user, operator, software, and other systems for correctness, consistency, completeness, accuracy, and testability. The task criteria are as follows:</p> <p>Correctness - Validate the external and internal software interface code in the context of systems requirements.</p> <p>Consistency - Verify that the interface code is consistent between source code components and to external interfaces (i.e., hardware, user, operator, and other software).</p> <p>Completeness - Verify that each interface is described and includes data format and performance criteria (e.g., timing, bandwidth, accuracy, safety, and security).</p> <p>Accuracy - Verify that each interface provides information with the required accuracy.</p> <p>Testability - Verify that there are objective acceptance criteria for validating the interface code.</p>
System Test Case Analysis	5.4	<p>Verify that the System Test Cases comply with project defined test document purpose, format, and content. Validate that the System Test Cases satisfy the criteria in the System Test Plan.</p>
Software FQT Case Analysis	5.5	<p>Verify that the Software FQT Cases comply with project defined test document purpose, format, and content. Validate that the Software FQT Cases satisfy the criteria in the Software FQT Plan.</p>
Software Integration Test Case Analysis	5.6	<p>Verify that the Software Integration Test Cases comply with project defined test document purpose, format, and content. Validate that the Software Integration Test Cases satisfy the criteria in the Software Integration Test Plan.</p>

CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE


 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Acceptance Test Case Analysis	5.7	Verify that the Acceptance Test Cases comply with project defined test document purpose, format, and content. Validate that the Acceptance Test Cases satisfy the criteria in the Acceptance Test Plan.
Software Integration Test Procedure Analysis	5.8	Verify that the Software Integration Test Procedures comply with project defined test document purpose, format, and content. Validate that the Software Integration Test Procedures satisfy the criteria in the Software Integration Test Plan.
Software Integration Test Results Analysis	5.9	Use the developer's integration test results to verify that the software components are integrated correctly. Verify that the test results trace to the test criteria established by the test traceability in the test planning documents. Document discrepancies between actual and expected results
Component Test Case Analysis	5.10	Verify that the Component Test Cases comply with project defined test document purpose, format, and content. Validate that the Component Test Cases satisfy the criteria in the Component Test Plan.
System Test Procedure Analysis	5.11	Verify that the System Test Procedures comply with project defined test document purpose, format, and content. Validate that the System Test Procedures satisfy the criteria in the System Test Plan.
Software FQT Procedure Analysis	5.12	Verify that the Software FQT Procedures comply with project defined test document purpose, format, and content. Validate that the Software FQT Procedures satisfy the criteria in the Software FQT Plan.

Test Phase 6.0

Traceability Analysis - Test	6.1	Analyze relationships in the Test Plans, Designs, Cases, and Procedures for correctness and completeness. For correctness, verify that there is a valid relationship between the Test Plans, Designs, Cases, and Procedures. For completeness, verify that all Test Procedures are traceable to the Test Plans. Perform this tracing only for test types and documents subject to IV&V test analysis.
Regression Test Analysis	6.2	Determine the extent of the testing that must be repeated when changes are made to any previously examined software products. Assess the nature of the changes to determine potential ripple of side effects and impacts on other aspects of the system.
Simulation Analysis	6.3	Analyze the simulation for correctness, accuracy and completeness. For correctness, verify that the simulation satisfies the simulation and system requirements For accuracy, validate that the modeled physical phenomena conform to physical laws and that the simulation accurately represents the system environment. For completeness, verify that the simulation has all the functionality necessary to perform the intended testing.
System Test Results Analysis	6.4	Use the developer's system test results to validate that the software satisfies the system requirements. Verify that the test results trace to the test criteria established by the test traceability in the test planning documents. Document discrepancies between actual and expected results.

CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Software FQT Results Analysis	6.5	Use the developer's software FQT test results to validate that the software satisfies the software requirements. Verify that the test results trace to the test criteria established by the test traceability in the test planning documents. Document discrepancies between actual and expected results.
-------------------------------	-----	---


Operations and Maintenance Phases 7.0

Operating Procedure Evaluation	7.1	Verify that the operating procedures are consistent with the user documentation and conform to the system requirements.
Anomaly Evaluation	7.2	Evaluate the effect of software operation anomalies.
Migration Assessment	7.3	Assess whether the software requirements and implementation address 1) specific migration requirements, 2) migration tools, 3) conversion of software products and data, 4) software archiving, 5) support for the prior environment, and 6) user notification.
Retirement Assessment	7.4	For software retirement, assess whether the installation package addresses: 1) software support, 2) impact on existing systems and databases, 3) software archiving, 4) transition to a new software product, and 5) user notification.


Optional Tasks 8.0

Acceptance Test Design Analysis	8.1	Verify that the Acceptance Test Designs comply with project defined test document purpose, format, and content. Validate that the Acceptance Test Designs satisfy the criteria in the Acceptance Test Plan.
Acceptance Test Procedure Analysis	8.2	Verify that the Acceptance Test Procedures comply with project defined test document purpose, format, and content. Validate that the Acceptance Test Procedures satisfy the criteria in the Acceptance Test Plan.
Acceptance Test Results Analysis	8.3	Use the developer's acceptance test results to validate that the software satisfies the test acceptance criteria. Validate that the test results trace to the test criteria established by the test traceability in the test planning documents. Document discrepancies between actual and expected results.
Algorithm Analysis	8.4	Verify the correct implementation of algorithms, equations, mathematical formulations, or expressions. Re-derive any significant algorithms and equations from basic principles and theories. Compare against established references or proven past historical data. Validate the algorithms, equations, mathematical formulations, or expressions with respect to the system and software requirements. Ensure that the algorithms and equations are appropriate for the problem solution. Validate the correctness of any constraints or limitations such as rounding, truncation, expression simplifications, best fit estimations, and non-linear solutions imposed by the algorithms and equations.
Audit Support	8.5	[Describe the audit support to be provided]


CHECK THE MASTER LIST at <http://ims.ivv.nasa.gov/>
VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Component Test Design Analysis	8.6	Verify that the Component Test Designs comply with project defined test document purpose, format, and content. Validate that the Component Test Designs satisfy the criteria in the Software Component Test Plan.
Component Test Procedure Analysis	8.7	Verify that the Component Test Procedures comply with project defined test document purpose, format, and content. Validate that the Component Test Procedures satisfy the criteria in the Component Test Plan.
Component Test Results Analysis	8.8	Use the developer's component test results to validate that the software satisfies the test acceptance criteria. Validate that the test results trace to the test criteria established by the test traceability in the test planning documents. Document discrepancies between actual and expected results.
Configuration Management Assessment	8.9	Verify that the Configuration Management process is complete and adequate. Verify that there is a process for describing the software product functionality, tracking program versions, and managing changes. Verify that the configuration management process is adequate for the development complexity, software and system size, software integrity level, project plans, and user needs.
Disaster Recovery Plan	8.10	Verify that the disaster recovery plan is adequate to restore critical operation of the system in the case of an extended system outage. The disaster recovery plan should include the following: <ul style="list-style-type: none"> 1) Identification of the disaster recovery team and a contact list. 2) Recovery operation procedures. 3) Procedure for establishing an alternative site including voice and data communications, mail, and support equipment. 4) Plans for replacement of computer equipment. 5) Establishment of a system backup schedule. 6) Procedures for storage and retrieval of software, data, documentation, and vital records off-site. 7) Logistics of moving staff, data, documentation, etc.
Feasibility Study Evaluation	8.11	Verify that the feasibility study is correct, accurate, and complete. Validate that all logical and physical assumptions (e.g., physical models, business rules, and logical processes), constraints, and user requirements are satisfied.
Independent Testing	8.12	[Describe in detail independent testing to be performed]
Operational Evaluation	8.13	Assess the deployment readiness and operational readiness of the software. Operational Evaluation shall include examining the results of operational tests, audit reviews, and anomaly reports. Verify that the software is at a suitable point of correctness for deployment and correct for site specific configurations.

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

Performance Monitoring	8.14	<p>Collect information on the performance of software under operational conditions. Determine whether system and software performance requirements are satisfied. Performance monitoring shall include evaluation of the following items as applicable to the system:</p> <ol style="list-style-type: none"> 1) Database transaction rates to determine the need to reorganize or re-index the database. 2) CPU performance monitoring for load balancing. 3) Direct access storage utilization. 4) Network traffic to ensure adequate bandwidth. 5) Critical outputs of a system (e.g., scheduled frequency, expected range of values, scheduled system reports, reports of events).
Project Management Oversight Support	8.15	<p>Assess project development status for technical and management issues, risks, and problems. Coordinate oversight assessment with the acquirer and development organization. Evaluate project plans, schedules, development processes, and status. Collect, analyze, and report on key project metrics.</p>
Security Assessment	8.16	<p>Review the system owner's definition of an acceptable level of security risk. Analyze the system from a security perspective, and ensure that potential security risks with respect to confidentiality, integrity, availability, and accountability have been identified. Include an assessment of the sensitivity of the information/data to be processed. Analyze security risks introduced by the system itself as well as those associated with the environment with which the system interfaces.</p>
Software FQT Design Analysis	8.17	<p>Verify that the Software FQT Designs comply with project defined test document purpose, format, and content. Validate that the Software FQT Designs satisfy the criteria in the Software FQT Plan.</p>
Software Integration Test Design Analysis	8.18	<p>Verify that the Software Integration Test Designs comply with project defined test document purpose, format, and content. Validate that the Software Integration Test Designs satisfy the criteria in the Software Integration Test Plan.</p>
System Test Design Analysis	8.19	<p>Verify that the System Test Designs comply with project defined test document purpose, format, and content. Validate that the System Test Designs satisfy the criteria in the System Test Plan.</p>
Training Documentation Evaluation	8.20	<p>Evaluate the training materials and procedures for completeness, correctness, readability, and effectiveness.</p>
User Documentation Evaluation	8.21	<p>Evaluate the user documentation for its completeness, correctness, and consistency with respect to requirements for user interface and for any functionality that can be invoked by the user. The review of the user documentation for its readability and effectiveness shall include representative end users who are unfamiliar with the software. Employ the user documentation in planning and acceptance test that is representative of the operational environment.</p>

 Independent Verification & Validation Facility	Independent Verification and Validation	IVV 09-1 Revision: H Effective Date: May 14, 2004
--	--	--

7.0 Metrics

There are no metrics for the IVV 09-1 system level procedure.

8.0 Records

All forms, reports, and documents identified in this procedure follow the record guidelines specified in IVV 09-4, Section 8.0.

Note: For records retention, refer to new NASA Directives System as a NASA Procedural Requirement (NPR) for NASA Records Retention Schedules (NRRS), NPR 1441.1.